Abgabe: 11.11.2008 (vor der Vorlesung)

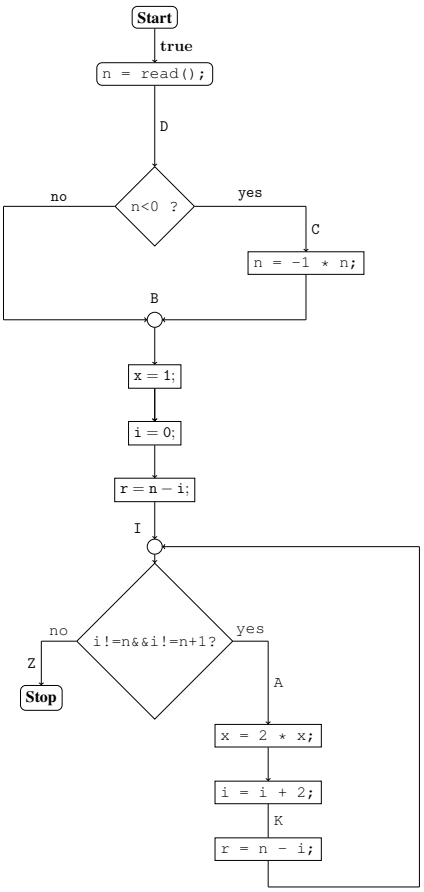
Aufgabe 4.1 (H) Terminierung

Zeigen Sie, dass folgendes MiniJava-Programm stets terminiert.

```
int i, x, n;
n = read();
if (n < 0)
  n = -1 * n;
x = 1;
i = 0;
while (i != n && i != n+1) {
  x = 2 * x;
  i = i + 2;
}
```

Lösungsvorschlag 4.1

Wir erstellen zunächst das instrumentierte Kontrollfluß-Diagramm:



Wir setzen

$$\mathtt{Z} :\equiv \mathtt{true}.$$

Wir raten

$$\mathtt{I} :\equiv \mathtt{r} = \mathtt{n} - \mathtt{i} \wedge \mathtt{i} \leq \mathtt{n} + 1.$$

Es ergibt sich

$$\begin{aligned}\mathbf{WP}[\![\mathtt{r} = \mathtt{n} - \mathtt{i};]\!](\mathtt{I}) &\equiv \mathtt{n} - \mathtt{i} = \mathtt{n} - \mathtt{i} \wedge \mathtt{i} \leq \mathtt{n} + 1 \\ &\equiv \mathtt{i} \leq \mathtt{n} + 1 \\ &\Leftarrow \mathtt{i} \leq \mathtt{n} + 1 \wedge \mathtt{n} - \mathtt{i} < \mathtt{r} \\ &\Leftarrow \mathtt{i} \leq \mathtt{n} + 1 \wedge \mathtt{n} - \mathtt{i} + 1 < \mathtt{r} \\ &\equiv: \mathtt{K}\end{aligned}$$

$$\begin{aligned}\mathbf{WP}[\![\mathtt{x} = 2 * \mathtt{x}; \mathtt{i} = \mathtt{i} + 2;]\!](\mathtt{K}) &\equiv \mathtt{i} + 2 \leq \mathtt{n} + 1 \wedge \mathtt{n} - \mathtt{i} - 1 < \mathtt{r} \\ &\equiv \mathtt{i} < \mathtt{n} \wedge \mathtt{n} - \mathtt{i} - 1 < \mathtt{r} \\ &\equiv: \mathtt{A}\end{aligned}$$

Wir überprüfen:

$$\begin{aligned}&\mathbf{WP}[\![\mathtt{i}! = \mathtt{n} \;\&\&\; \mathtt{i}! = \mathtt{n} + 1?]\!](\mathtt{Z}, \mathtt{A}) \\ &\equiv (\mathtt{i} \in \{\mathtt{n}, \mathtt{n} + 1\}) \vee (\mathtt{i} \notin \{\mathtt{n}, \mathtt{n} + 1\} \wedge \mathtt{i} < \mathtt{n} \wedge \mathtt{n} - \mathtt{i} - 1 < \mathtt{r}) \\ &\equiv \mathtt{i} \in \{\mathtt{n}, \mathtt{n} + 1\} \vee (\mathtt{i} < \mathtt{n} \wedge \mathtt{n} - \mathtt{i} - 1 < \mathtt{r}) \\ &\Leftarrow \mathtt{i} \in \{\mathtt{n}, \mathtt{n} + 1\} \vee (\mathtt{i} < \mathtt{n} \wedge \mathtt{n} - \mathtt{i} = \mathtt{r}) \\ &\Leftarrow \mathtt{i} \leq \mathtt{n} + 1 \wedge \mathtt{n} - \mathtt{i} = \mathtt{r} \\ &\equiv \mathtt{I}\end{aligned}$$

Es ergibt sich

$$\begin{aligned}\mathbf{WP}[\![\mathtt{x} = 1; \mathtt{i} = 0; \mathtt{r} = \mathtt{n} - \mathtt{i};]\!](\mathtt{I}) &\equiv \mathtt{n} - 0 = \mathtt{n} - 0 \wedge 0 \leq \mathtt{n} + 1 \\ &\Leftarrow 0 \leq \mathtt{n} \\ &\equiv: \mathtt{B}\end{aligned}$$

$$\begin{aligned}\mathbf{WP}[\![\mathtt{n} = -1 * \mathtt{n};]\!](\mathtt{B}) &\equiv \mathtt{n} \leq 0 \\ &\equiv: \mathtt{C}\end{aligned}$$

$$\begin{aligned}\mathbf{WP}[\![\mathtt{n} < 0]\!](\mathtt{B}, \mathtt{C}) &\equiv (\mathtt{n} \geq 0 \wedge \mathtt{n} \geq 0) \vee (\mathtt{n} < 0 \wedge \mathtt{n} \leq 0) \\ &\equiv: \mathtt{n} \geq 0 \vee \mathtt{n} < 0 \\ &\equiv: \mathbf{true} \\ &\equiv: \mathtt{D}\end{aligned}$$

Wichtig: Es gelten folgende Aussagen:

a) $\mathtt{A} \Rightarrow \mathtt{r} > 0$

b) $\mathtt{K} \Rightarrow \mathtt{n} - \mathtt{i} < \mathtt{r}$

Aufgabe 4.2 (H) Binomialkoeffizient

Gegeben sei folgendes Code-Fragment:

```
int i,k,n,b;

n = read();
k = read();
if (0 <= k && k <= n) {
  b = 1;
  i = 1;
  while (i <= k) {
    b = b * (n + 1 - i);
    b = b / i;
    i = i + 1;
  }
} else {
  b = 0;
}
write(b);
```

Zeigen Sie, dass das gegebene Code-Fragment den Binomialkoeffizienten $\binom{\mathtt{n}}{\mathtt{k}}$ berechnet und ausgibt. Dieser ist durch

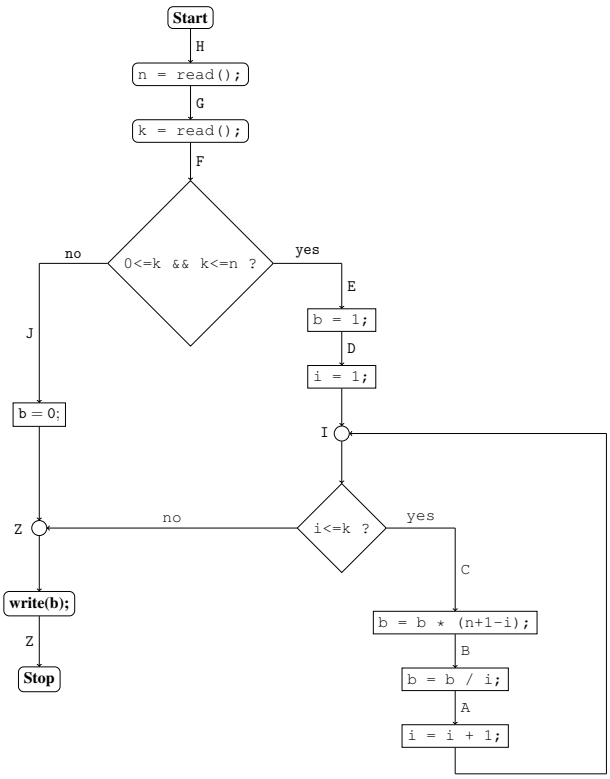
$$\binom{n}{k} = \begin{cases} \frac{n!}{(n-k)!\cdot k!} & \text{falls } 0 \leq k \leq n \text{ gilt} \\ 0 & \text{andernfalls} \end{cases}, \qquad n,\ k \in \mathbb{Z}$$

gegeben. Am Stop-Knoten muss also die Zusicherung $\mathtt{b} = \binom{\mathtt{n}}{\mathtt{k}}$ gelten.

Hinweis: Ein Korrektheitsbeweis sichert auch, dass, unmittelbar vor der Ausführung der Anweisung $\mathtt{b} = \mathtt{b}/\mathtt{i};$, $\mathtt{b}$ immer durch $\mathtt{i}$ teilbar ist, d.h. es gilt $\frac{\mathtt{b}}{\mathtt{i}} \in \mathbb{Z}$ unmittelbar vor der Anweisung.

Lösungsvorschlag 4.2

Wir erstellen zunächst das Kontrollfluß-Diagramm:



Wir setzen

$$\mathtt{Z} :\equiv \mathtt{b} = \binom{\mathtt{n}}{\mathtt{k}}.$$

Wir raten

$$\mathtt{I} :\equiv 1 \leq \mathtt{i} \leq \mathtt{k}+1 \leq \mathtt{n}+1 \wedge \mathtt{b} = \binom{\mathtt{n}}{\mathtt{i}-1}.$$

Es folgt:

$$
\begin{aligned}
\mathbf{WP}[\![\mathtt{i} = \mathtt{i}+1;]\!](\mathtt{I}) &\equiv 1 \le \mathtt{i}+1 \le \mathtt{k}+1 \le \mathtt{n}+1 \wedge \mathtt{b} = \binom{\mathtt{n}}{\mathtt{i}} \\
&\equiv 0 \le \mathtt{i} \le \mathtt{k} \le \mathtt{n} \wedge \mathtt{b} = \binom{\mathtt{n}}{\mathtt{i}} \equiv: \mathtt{A} \\
\mathbf{WP}[\![\mathtt{b} = \mathtt{b}/\mathtt{i};]\!](\mathtt{A}) &\equiv 0 \le \mathtt{i} \le \mathtt{k} \le \mathtt{n} \wedge \mathtt{b}\ \mathbf{div}\ \mathtt{i} = \binom{\mathtt{n}}{\mathtt{i}} \Leftarrow 1 \le \mathtt{i} \le \mathtt{k} \le \mathtt{n} \wedge \frac{\mathtt{b}}{\mathtt{i}} = \binom{\mathtt{n}}{\mathtt{i}} \equiv: \mathtt{B} \\
\mathbf{WP}[\![\mathtt{b} = \mathtt{b} * (\mathtt{n}+1-\mathtt{i});]\!](\mathtt{B}) &\equiv 1 \le \mathtt{i} \le \mathtt{k} \le \mathtt{n} \wedge \frac{\mathtt{b} \cdot (\mathtt{n}+1-\mathtt{i})}{\mathtt{i}} = \binom{\mathtt{n}}{\mathtt{i}} = \frac{\mathtt{n}!}{(\mathtt{n}-\mathtt{i})! \cdot \mathtt{i}!} \\
&\equiv 1 \le \mathtt{i} \le \mathtt{k} \le \mathtt{n} \wedge \mathtt{b} = \frac{\mathtt{n}! \cdot \mathtt{i}}{(\mathtt{n}-\mathtt{i})! \cdot \mathtt{i}! \cdot (\mathtt{n}-(\mathtt{i}-1))} \\
&\equiv 1 \le \mathtt{i} \le \mathtt{k} \le \mathtt{n} \wedge \mathtt{b} = \frac{\mathtt{n}!}{(\mathtt{n}-(\mathtt{i}-1))! \cdot (\mathtt{i}-1)!} = \binom{\mathtt{n}}{\mathtt{i}-1} \equiv: \mathtt{C} \\
\mathbf{WP}[\![\mathtt{i} <= \mathtt{k}]\!](\mathtt{Z}, \mathtt{C}) &\equiv (\mathtt{i} > \mathtt{k} \wedge \mathtt{b} = \binom{\mathtt{n}}{\mathtt{k}}) \vee (\mathtt{i} \le \mathtt{k} \wedge 1 \le \mathtt{i} \le \mathtt{k} \le \mathtt{n} \wedge \mathtt{b} = \binom{\mathtt{n}}{\mathtt{i}-1}) \\
&\Leftarrow (\mathtt{i} > \mathtt{k} \wedge 1 \le \mathtt{i} \le \mathtt{k}+1 \le \mathtt{n}+1 \wedge \mathtt{b} = \binom{\mathtt{n}}{\mathtt{i}-1}) \\
&\qquad \vee (\mathtt{i} \le \mathtt{k} \wedge 1 \le \mathtt{i} \le \mathtt{k}+1 \le \mathtt{n}+1 \wedge \mathtt{b} = \binom{\mathtt{n}}{\mathtt{i}-1}) \\
&\equiv (\mathtt{i} > \mathtt{k} \vee \mathtt{i} \le \mathtt{k}) \wedge 1 \le \mathtt{i} \le \mathtt{k}+1 \le \mathtt{n}+1 \wedge \mathtt{b} = \binom{\mathtt{n}}{\mathtt{i}-1}) \\
&\equiv 1 \le \mathtt{i} \le \mathtt{k}+1 \le \mathtt{n}+1 \wedge \mathtt{b} = \binom{\mathtt{n}}{\mathtt{i}-1}) \\
&\equiv \mathtt{I} \\
\mathbf{WP}[\![\mathtt{i} = 1;]\!](\mathtt{I}) &\equiv 0 \le \mathtt{k} \le \mathtt{n} \wedge \mathtt{b} = \binom{\mathtt{n}}{0} = 1 \equiv: \mathtt{D} \\
\mathbf{WP}[\![\mathtt{b} = 1;]\!](\mathtt{D}) &\equiv 0 \le \mathtt{k} \le \mathtt{n} \equiv: \mathtt{E} \\
\mathbf{WP}[\![\mathtt{b} = 0;]\!](\mathtt{Z}) &\equiv 0 = \binom{\mathtt{n}}{\mathtt{k}} \equiv: \mathtt{J} \\
\mathbf{WP}[\![0 <= \mathtt{k}\&\&\mathtt{k} <= \mathtt{n}]\!](\mathtt{J}, \mathtt{E}) &\equiv (\neg(0 \le \mathtt{k} \wedge \mathtt{k} \le \mathtt{n}) \wedge 0 = \binom{\mathtt{n}}{\mathtt{k}}) \vee (0 \le \mathtt{k} \wedge \mathtt{k} \le \mathtt{n} \wedge 0 \le \mathtt{k} \le \mathtt{n}) \\
&\equiv (\neg(0 \le \mathtt{k} \wedge \mathtt{k} \le \mathtt{n})) \vee (0 \le \mathtt{k} \wedge \mathtt{k} \le \mathtt{n}) \\
&\equiv \mathbf{true} \\
&\equiv: \mathtt{F} \\
\mathbf{WP}[\![\mathtt{k} = \mathtt{read}();]\!](\mathtt{F}) &\equiv \mathbf{true} \equiv: \mathtt{G} \\
\mathbf{WP}[\![\mathtt{k} = \mathtt{read}();]\!](\mathtt{G}) &\equiv \mathbf{true} \equiv: \mathtt{H}
\end{aligned}
$$

Aufgabe 4.3 (P) Adaption

Nehmen Sie an, dass folgendes Tripel für die nicht näher spezifizierte Prozedur `f()` gültig sei:

$$\{\mathtt{x} = l_\mathtt{x} \wedge \mathtt{y} = l_\mathtt{y} \wedge l_\mathtt{x} \leq l_\mathtt{y}\} \quad \mathtt{f}(); \quad \{\mathtt{z} = 12 \cdot l_\mathtt{x} - 3 \cdot l_\mathtt{y} \wedge \mathtt{y} = l_\mathtt{y}\} \tag{1}$$

a) Wie ist der Wert der Variablen `z` nach Ausführung der Prozedur `f()`, wenn vor Ausführung `x` den Wert 1 und `y` den Wert 0 hat?

b) Wie ist der Wert der Variablen `z` nach Ausführung der Prozedur `f()`, wenn vor Ausführung `x` den Wert 1 und `y` den Wert 2 hat?

c) Wie ist der Wert der Variablen `x` nach Ausführung der Prozedur `f()`, wenn vor Ausführung `x` den Wert 7 hat?

d) Welche der folgenden Tripel sind gültig? Begründen Sie Ihre Antwort!

Für ein Tripel dessen Gültigkeit nicht durch die Gültigkeit des Tripels in (1) impliziert wird, ist ein Gegenbeispiel anzugeben. D.h. es ist eine Prozedur `f()` anzugeben für die das Tripel in (1) erfüllt aber das Tripel in der Teilaufgabe nicht erfüllt ist. Um zu zeigen, dass ein Tripel nicht gilt sind lediglich Werte für `x` und `y` zu benennen für die das Tripel nicht gilt.

i) $\{\mathtt{x} = l_\mathtt{x} = \mathtt{y} = l_\mathtt{y}\} \quad \mathtt{f}(); \quad \{\mathtt{z} = 12 \cdot l_\mathtt{x} - 3 \cdot l_\mathtt{y} \wedge \mathtt{y} = l_\mathtt{y}\}$

ii) $\{\mathtt{x} = l_\mathtt{x} \wedge \mathtt{y} = l_\mathtt{y}\} \quad \mathtt{f}(); \quad \{\mathtt{z} = 12 \cdot l_\mathtt{x} - 3 \cdot l_\mathtt{y} \wedge \mathtt{y} = l_\mathtt{y}\}$

iii) $\{\mathtt{x} = l_\mathtt{x} \wedge \mathtt{y} = l_\mathtt{y} \wedge l_x \leq l_\mathtt{y}\} \quad \mathtt{f}(); \quad \{\mathtt{x} = l_\mathtt{x} \Rightarrow (\mathtt{z} = 12 \cdot l_\mathtt{x} - 3 \cdot l_\mathtt{y} \wedge \mathtt{y} = l_\mathtt{y})\}$

iv) $\{\mathtt{x} = l_\mathtt{x} \wedge \mathtt{y} = l_\mathtt{y} + l_\mathtt{x} + 1 \wedge -1 \leq l_\mathtt{y}\} \quad \mathtt{f}(); \quad \{\mathtt{z} = 9 \cdot l_\mathtt{x} - 3 \cdot l_\mathtt{y} - 3 \wedge \mathtt{y} = l_\mathtt{y} + l_\mathtt{x} + 1\}$

v) $\{0 < \mathtt{x} \wedge \mathtt{x} < \mathtt{y} \wedge \mathtt{y} < 0\} \quad \mathtt{f}(); \quad \{\mathtt{z} = 42\}$

e) Geben Sie eine Prozedur `f()` an, für die das Tripel

$$\{\text{true}\} \quad \mathtt{f}(); \quad \{\text{false}\}$$

erfüllt ist.

Lösungsvorschlag 4.3

a) Über diesen Fall macht das Tripel keine Aussage.

b) In diesem Fall hat `z` nach Ausführung der Prozedur den Wert 6.

c) Darüber macht das Tripel keine Aussage.

d) i) Es ist gültig, da es aus dem Tripel in (1) durch Verstärkung der Vorbedingung entstanden ist.

ii) Es ist nicht gültig. Gegenbeispiel: Die folgende Prozedur erfüllt das Tripel aus (1).

```
void f() {
  if (x<=y)
    z = 12 * x - 3 * y;
  else
    z = 12 * x - 3 * y - 1;
}
```

Das Tripel in der Aufgabenstellung wird jedoch aus folgendem Grund nicht erfüllt. Der Zustand $\sigma = \{\mathtt{x} \mapsto 1, \mathtt{y} \mapsto 0, \mathtt{z} \mapsto 0\}$ erfüllt die Vorbedingung (mit $l_{\mathtt{x}} = 1$ und $l_{\mathtt{y}} = 0$). Wird die Prozedur mit dem Zustand σ aufgerufen, so hat das Programm den Zustand $\sigma' = \{x \mapsto 1, y \mapsto 0, z \mapsto 11\}$ nach dem Prozeduraufruf. Der Zustand σ' erfüllt jedoch nicht die Nachbedingung (mit $l_{\mathtt{x}} = 1$ und $l_{\mathtt{y}} = 0$).

iii) Es ist gültig, da es aus dem Tripel in (1) durch Abschwächung der Nachbedingung entstanden ist.

iv) Es ist gültig, da es aus dem Tripel in (1) durch Substitution der logischen Variablen $l_{\mathtt{y}}$ durch den Term $l_{\mathtt{y}} + l_{\mathtt{x}} + 1$ entstanden ist. Wichtig dabei ist, dass der Term $l_{\mathtt{y}} + l_{\mathtt{x}} + 1$ nur logische Variablen enthält.

v) Gültig, da $\mathtt{0} < \mathtt{x} \wedge \mathtt{x} < \mathtt{y} \wedge \mathtt{y} < \mathtt{0} \equiv \mathbf{false}$ gilt.

e)
```
void f() {
  while(true);
}

// oder

void f() {
  f();
}
```

Aufgabe 4.4 (P) Prozeduren

Gegeben sei folgendes Code-Fragment:

```
int n, r;

public void fact() {
        /* Ihr Code */
}
```

a) Ergänzen Sie das Fragment zu einer **nicht-rekursiven** Prozedur für die das Tripel

$$\{\mathtt{n} = l \geq 0\} \quad \mathtt{fact}(); \quad \{\mathtt{r} = l!\}$$

gilt, wobei l eine logische Variable ist. Zur Erinnerung:

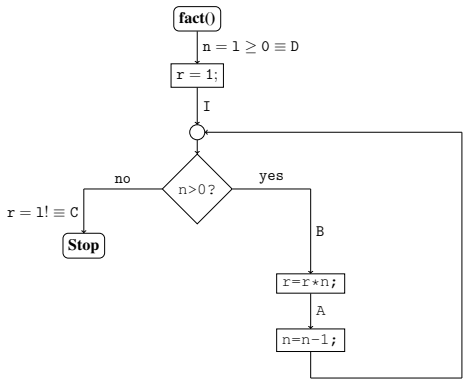
$$l! = \prod_{i=1}^{l} i = l \cdot (l-1) \cdot (l-2) \cdot \cdots \cdot 1, \qquad l \geq 0.$$

Im Speziellen gilt: $0! = 1$.

b) Zeigen Sie die Korrektheit Ihrer Implementierung!

Lösungsvorschlag 4.4

Wir erstellen zunächst das Kontrollfluß-Diagramm:



Wir raten

$$\mathtt{I} :\equiv \mathtt{r} = \frac{\mathtt{l}!}{\mathtt{n}!} \wedge \mathtt{l}, \mathtt{n} \geq 0.$$

Es ergibt sich

$$\begin{aligned} \mathbf{WP}[\![\mathtt{n} = \mathtt{n} - 1;]\!](\mathtt{I}) &\equiv \mathtt{r} = \frac{\mathtt{l}!}{(\mathtt{n}-1)!} \wedge \mathtt{n} > 0 \wedge \mathtt{l} \geq 0 \\ &\equiv: \mathtt{A} \end{aligned}$$

$$\begin{aligned} \mathbf{WP}[\![\mathtt{r} = \mathtt{r} * \mathtt{n};]\!](\mathtt{A}) &\equiv \mathtt{r} \cdot \mathtt{n} = \frac{\mathtt{l}!}{(\mathtt{n}-1)!} \wedge \mathtt{n} > 0 \wedge \mathtt{l} \geq 0 \\ &\Leftarrow \mathtt{r} = \frac{\mathtt{l}!}{\mathtt{n}!} \wedge \mathtt{n} > 0 \wedge \mathtt{l} \geq 0 \\ &\equiv: \mathtt{B} \end{aligned}$$

$$\begin{aligned}
\mathbf{WP}[\![\mathtt{n} > 0]\!](\mathtt{C},\mathtt{B}) &\equiv (\mathtt{n} \leq 0 \wedge \mathtt{r} = \mathtt{l}!) \vee (\mathtt{n} > 0 \wedge \mathtt{r} = \frac{\mathtt{l}!}{\mathtt{n}!} \wedge \mathtt{n} > 0 \wedge \mathtt{l} \geq 0) \\
&\Leftarrow (\mathtt{n} = 0 \wedge \mathtt{r} = \frac{\mathtt{l}!}{\mathtt{n}!} \wedge \mathtt{l} \geq 0) \vee (\mathtt{n} > 0 \wedge \mathtt{r} = \frac{\mathtt{l}!}{\mathtt{n}!} \wedge \mathtt{l} \geq 0) \\
&\equiv (\mathtt{n} = 0 \vee \mathtt{n} > 0) \wedge \mathtt{r} = \frac{\mathtt{l}!}{\mathtt{n}!} \wedge \mathtt{l} \geq 0 \\
&\equiv \mathtt{l}, \mathtt{n} \geq 0 \wedge \mathtt{r} = \frac{\mathtt{l}!}{\mathtt{n}!} \\
&\equiv \mathtt{I}
\end{aligned}$$

$$\begin{aligned}
\mathbf{WP}[\![\mathtt{r} = 1;]\!](\mathtt{I}) &\equiv \mathtt{l}, \mathtt{n} \geq 0 \wedge 1 = \frac{\mathtt{l}!}{\mathtt{n}!} \\
&\Leftarrow \mathtt{l} = \mathtt{n} \geq 0 \\
&\equiv \mathtt{D}
\end{aligned}$$